

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: CJ-1867

### REMARKS/ARGUMENTS

Reconsideration is respectfully requested.

The Abstract of the Disclosure has been added in this amendment in response to the Examiner's request during the Examiner interview on April 7, 2004. The Abstract contains about 150 words as required under the USPTO rules.

Claims 10-18 are pending in the present application before this amendment. By the present amendment, Claim 10 has been amended. No new matter has been added.

Claims 10-12 stand rejected under 35 U.S.C. § 103(a) as being obvious over "Fault-Tolerance for Communicating Multidatabase Transactions" (Kuhn94) in view of U.S. Patent No. 5,734,898 (He). The "et al." suffix, which may appear after a reference name, is omitted in this paper.

In response to the standing rejections to the claims, an Examiner interview was held on April 7, 2004 between the Applicant of the present application, Dr. Kuhn, the undersigned attorney of the record, and the Examiner. It was agreed during the Interview that the amended Claim 10 as shown in the Amendment "read[s] over the current rejection" as this has been summarized in the Interview Summary dated April 16, 2004. Accordingly, Applicant respectfully requests allowance of Claim 10 and the remaining pending claims that depend from Claim 10. Applicant thanks the Examiner for his time and attention given during the Interview.

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: CJ-1867

Additionally to the Interview, Applicant provides the following remarks.

Applicant respectfully agrees with the Examiner's statement in the Office Action of July 31, 2002, page 3, that Kuhn94 does not teach updateable objects and transactional blocking read of updateable objects.

In response to the Examiner's assertion in the same Office Action that He allegedly teaches the claimed updateable objects, Applicant in the Amendment filed January 31, 2003 has asserted that He fails to provide the updateable objects of the present invention because according to the present invention --at least some of the objects are updateable objects having a non-resettable logical time stamp and capable of storing data, wherein the updateable objects are coordinated by an optimistic concurrency control without utilizing explicit locks on the objects and further wherein the data of the updateable object is writeable on distributed peer nodes--. The bolded and underline features of the invention have now been incorporated into the Claim 10 of this Amendment.

For support of this amendment, Applicant respectfully reasserts the remarks offered in pages 4-5 of the Amendment filed on January 31, 2003.

Further, updateable objects of the presently claimed system are coordinated by means of an **optimistic concurrency control**, whereas He uses pessimistic locking as means to control concurrency. It is further emphasized that the presently claimed system **does not use explicit locks on objects at all**. Therefore, the claimed updateable objects have a different meaning when they are compared with those (i.e., "the objects being updated") taught in He. For instance, "the objects being updated" of He do not serve for the communication between different nodes.

According to the presently claimed invention, transactions may write (i.e. **physically store data on the disk**) an **updateable object's data on many distributed nodes**—not only on the one server site, as at He. In the present "CoKe" system, different consistency models are supported for objects (e.g., via the selectable distribution strategies such as the basic

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: C J-1867

strategy plus flags as recited in Claim 11).

Therefore, as updateable objects have a specific meaning in the CcKe system, and specific semantics have been developed for the meaning of transactional read updateable objects in CoKe.

For this reason, a logical time stamp is introduced for updateable objects, and the time stamps have other purposes than He's version number:

(1) The logical time stamp is never reset—not even after a cache reload of the object (In the present CoKe system, the logical time stamp is an inseparable part of the updateable object). The read API must specify a logical time stamp (so-called “logical read time stamp”) that has nothing in common with the logical time stamp of the very object in the respective local cache. The semantics of the logical read time stamp are: “read the next value of the object with the condition that the logical read time of this object must be greater than the specified logical read time stamp.”

(2) Depending on the consistency model used for the object, the CoKe read-API may or may not return a value that is the most recent value. Only the above-described condition must be met. In contrast, the server in He's system always returns the most recent value of the object.

Accordingly, neither Kuhn94 nor He, whether they are taken individually or together, teaches the claimed invention as presented in the amended Claim 10.

Furthermore, Claim 10 has been amended to recite —a peer-to-peer coordination system—. Applicant has already pointed out this aspect of the invention in the Amendment filed January 31, 2003, that:

...the presently claimed coordination system is completely different from He. In particular, the claimed coordination system is a peer-to-peer system, but the system taught in He is a client/server system to the contrary. Accordingly, the server of the presently claimed system runs on each node.

Therefore, the client/server system of He is quite substantially different from the peer-to-peer coordination of the presently claimed invention.

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: C J-1867

Indeed, the claimed peer-to-peer system is considered a separate field of technology as opposed to the client/server system such as He. A conventional client/server system requires a central computer to take the role of a central server in the network of computers. When the central server fails, the network would also fail--causing the "availability" problem in the network. The central server can only handle a limited number of client computers--this may lead to the "performance" problem in the network when a central server cannot handle a large number of clients. Further, a client computer cannot directly communicate to another client computer, unless a central server is available to direct such communication--this dependency on the central server also leads to the "performance" problems in the network.

In contradistinction, every computer in the claimed peer-to-peer system is a "peer" computer to each other. A computer in the claimed peer-to-peer system should be considered as being capable of communicating autonomously and directly with each other. In this regard alone, the claimed peer-to-peer system espouses a completely different way to organize a distributed network when compared to the conventional client/server system. CoKe, for example, is a technical solution on how to realize this peer-to-peer communication and in avoiding all of the problems associated with a conventional client/server system described above.

On this ground, Applicant respectfully notes that combining He with other references is considered to be improper at least since He relates to a different field of technological endeavor, i.e., He is not analogous to the presently

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: C J-1867

claimed invention.

Furthermore, because of this quite substantial difference in the system employed in He and the system of the presently claimed invention, there is no "reasonable expectation of success" as required in the MPEP §2143.02 even if He is combined with other references such as Kuhn94.

In the Specification pages 2-3 the conventional client/server architectures naturally have a hierarchical structure that distinguishes programs into either a client component or a server component, both of which have to be implemented by a programmer. Thus, in large applications, this hierarchical structure leads to performance problems.

Thus, even if the teachings of a conventional client/server system of He is combined with Kuhn94, the Office Action fails to provide that there is a reasonable success can be expected. Applicant respectfully notes that the Examiner bears the initial burden of factually supporting a prima facie conclusion of obviousness (MPEP §2141), one of which criteria is to show a reasonable expectation of success when the teachings of references are combined.

At least for the reasons above as already discussed and as agreed upon in the previous Examiner interview, Claim 10, as amended, reads over the rejections, and Applicant respectfully submits that Claim 10 is considered to be in condition for allowance. An indication thereof is respectfully requested.

Additionally, Applicant respectfully provides the support for each of the added limitations of Claim 10 below as requested by the Examiner in the interview.

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: C J-1867

The support for the --peer-to-peer system-- is found at least in the Specification page 3, line 3; page 8, line 13, page 8, line 17; page 12, line 13; page 14, line 5; page 9, line 8; page 18, line 1; page 20, line 16; and page 24, the last line.

The support for the --non-resettable logical time stamp-- is found at least in the Specification page 59, line 22.

The support for the --objects are capable of storing date-- is found at least in the Specification page 50, the paragraph at line 12 and page 43, the paragraph at line 23.

The support for the --updateable object-- is found at least in the Specification page 11, line 16; page 18, line 3; and page 21, line 1.

The support for the --optimistic concurrency control without utilizing explicit locks on the objects-- is found at least in the Specification page 14, the paragraph at line 10; page 40, the paragraph at line 12; page 39, line 8; and page 58, line 10 to page 60, line 13.

The support for --objects data are writeable on distributed nodes-- is found at least in the Specification page 39, the paragraph at line 3.

The support for the --logical time stamp-- , for example, the transactional blocking read of updateable objects using the logical time stamp, is found at least in the Specification page 44, the paragraph at line 5; and page 50, the last line to page 53, line 10.

Furthermore, an example for the purposes of describing the role of the logical time stamp in the updateable object for the optimistic concurrency

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: C J-1867

control is provided below. The Examiner in the interview requested such explanation. Although the below example description is well supported in and consistent with the present Specification, it is provided only as an example, and the scope of the presently claimed invention shall not be limited to or affected by the below example description. No intent to narrow the scope of the presently claimed invention should be inferred by the following example description.

Each object may have a non-resettable logical time stamp, which is stored with the object. The non-resettable logical time stamp may be considered as an inseparable part of the object. It is stored jointly with the value of the object, i.e. each value of an object relates to a logical time stamp.

After creation of the object, its logical time stamp may have a value equal to 0.

Every time when a value is written into the object (by a successfully committed transaction), its logical time stamp is incremented by 1. Therefore, the logical time stamp may be understood as having the meaning of a "value counter."

For a CONST object, the logical time stamp can only have one of two values, namely 0 (i.e., the object is still undefined) or 1 (i.e., the object is defined--it has been written once). For a VAR object (which may include an updateable object), the logical time stamp can take the values 0, 1, 2, 3 ...

One of the reasons why a logical time stamp is needed is to implement the blocking read of an updateable object, for example:

- (1) The `cobj_trans_read` request takes as argument a read time stamp (called "Time2"-- see the Specification page 50, the last line);
- (2) CoKe checks, if the logical time stamp of the object is larger than the given read time stamp; if yes, the object value is returned to the `cobj_trans_read` request; if no, the `cobj_trans_read` request blocks until the object becomes sufficiently defined (i.e., has been assigned so many values that its logical time stamp is now larger than the read time stamp);

Other reasons why the logical time stamp is needed is to implement optimistic concurrency control, for example:

- (1) If the `cobj_trans_read` returns a value (i.e. the condition

Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: CJ-1867

concerning that the read time stamp must be larger than the logical time stamp (see above) is fulfilled), then CoKe remembers the logical time stamp assigned to the returned read value;

(2) CoKe uses an optimistic concurrency control protocol; this means that there are no locks set on the object, if it is read (as is done in pessimistic concurrency control); this means that at commitment time CoKe must check, if the read value is still valid (because another transaction could have written the object meanwhile so that the read value is no longer the same as the current value of the object).

This check may work as follows at a commitment time:

- (1) In the first phase of the commitment, CoKe tries to get the main copy of the object (using the protocols of the strategy manager);
- (2) If the main copy is here (more precisely: if all main copies of all objects used in the transaction are here = second phase of this commitment), CoKe compares the logical time stamp of the object with the logical timestamp of the read value;
- (3) If both timestamps equal, everything is ok; otherwise the transaction cannot commit successfully.

For the reasons set forth above and further in view of the agreement of the Examiner interview of April 7, 2004, Applicant respectfully submits that Claims 10-18 pending in this application are in condition for allowance over the cited references. This amendment is considered to be responsive to all points raised in the Office Action. Accordingly, Applicant respectfully requests reconsideration and withdrawal of the outstanding rejections and earnestly

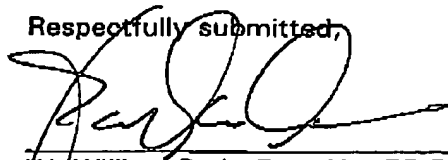


Application Serial No. 9/269,485  
Reply to Office Action of November 12, 2003

PATENT  
Docket: C J-1867

solicits an indication of allowable subject matter. Should the Examiner have any remaining questions or concerns, the Examiner is encouraged to contact the undersigned attorney by telephone to expeditiously resolve such concerns.

Respectfully submitted,



Dated: May 12, 2004

W. William Park, Reg. No. 55,523  
Ladas & Parry  
224 South Michigan Avenue  
Chicago, Illinois 60604  
(312) 427-1300